UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/629,284	07/29/2003	Arun Kwangil Iyengar	YOR920010663US1	5707

7590 12/04/2007

Ryan, Mason & Lewis, LLP
90 Forest Avenue
Locust Valley, NY 11560

EXAMINER	
DOAN, DUC T	

ART UNIT	PAPER NUMBER
2188	

MAIL DATE	DELIVERY MODE
12/04/2007	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

UNITED STATES PATENT AND TRADEMARK OFFICE

Commissioner for Patents
United States Patent and Trademark Office
P.O. Box 1450
Alexandria, VA 22313-1450
www.uspto.gov

BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

Application Number: 10/629,284
Filing Date: July 29, 2003
Appellant(s): IYENGAR, ARUN KWANGIL

William E. Lewis #39274
For Appellant

EXAMINER'S ANSWER

This is in response to the appeal brief filed 8/22/2007 appealing from the Office action mailed 3/2/2006.

(1) Real Party in interest

A statement identifying by name the real party interest is contained in the brief.

(2) Related Appeals and Interferences

The examiner is not aware of any related appeals, interferences, or judicial proceedings which will directly affect or be directly affected by or have a bearing on the Board's decision in the pending appeal.

(3) Status of Claims

The statement of the status of claims contained in the brief is correct.

(4) Status of Amendments After Final

No amendment after final has been filed.

(5) Summary of Claimed Subject Matter

The summary of claimed subject matter contained in the brief is correct.

(6) Grounds of Rejection to be Reviewed on Appeal

The appellant's statement of the grounds of rejection to be reviewed on appeal is correct.

(7) Claims Appendix

The copy of the appealed claims contained in the Appendix to the brief is correct.

(8) Evidence Relied Upon

US 6,915,307	Mattis et al	07-2005
US 2002/0032691	Rabbi et al	03-2002
US 6,804,761	Chen et al	10-2004
US 5,802,599	Cabrera et al	09-1998

US 2004/0172507	Garthwait, Alexander T.	09-2004

(9) Grounds of Rejection

The following ground(s) of rejection are applicable to the appealed claims:

Claim Rejections - 35 USC § 103

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

Claims 1-3,9,11-12,14-16,19-28 rejected under 35 U.S.C. 103(a) as being unpatentable over Rabii et al (US Pub 2002/0032691) and in view of Mattis et al (US 6915307).

As for claim 1, Rabbi describes a method of managing storage of objects of sizes smaller than a storage transfer unit in a computer system, comprising the steps of: maintaining a plurality of storage transfer units (segments) in a first storage medium organized by a quantity of free space in a storage transfer unit (Rabii's page 3, paragraphs 38,39; Fig 5); maintaining in a second storage medium a cache comprising a copy of at least one of said plurality of storage transfer units (Rabbi's Fig 10: #520 data buffers); The claim further recites in response to a request to store an object of a size less than a storage transfer unit : searching for a cached storage transfer unit with sufficient free space to store the object ; if no such cached storage transfer unit can be found, identifying an uncached storage transfer unit with sufficient free space to store the object and storing a copy of the identified storage transfer unit in the cache; and storing the object in the identified storage transfer unit by modifying at least one data structure in the cache and subsequently writing a cached copy of the storage transfer unit to the first storage medium. Although, Rabbi describes allocating new objects to a segment in a disk (Rabbi's paragraphs 83,84; Fig 5). Rabbi does not describe the claim's detail of other steps to handling an object's request. However, Mattis describes the situation of a client requesting for an information object and the object is not found in cache; the location of object is retrieved from a disk (Mattis's column 20, lines 21-30); the object is written into the allocated write aggregation buffer; the directory is updated; Subsequently, the buffer allocated write aggregation buffer is written back to an arena (Mattis's column 20, lines 40-65; Fig 10A); It would have been obvious to one of ordinary skill in the art at the time of invention to include the steps for storing an object as suggested by Mattis in Rabbi's system to assure the integrity of information objects (Mattis's column 21, lines 1-15).

As for claims 2-3, wherein the first storage medium comprises disk storage (claim 2; Rabbi's Fig 5); wherein the second storage medium comprises main memory (claim 3; Rabii's Fig 10).

As for claim 9, Rabii describes maintaining at least one list of storage transfer units (Rabii's Fig 5: #220-2 segment); maintaining at least one tail pointer to a plurality of contiguous unallocated storage transfer units (Rabii's Fig 5 a ring of allocated and unallocated segments);

wherein the step of identifying an uncached storage transfer unit further comprises the steps of searching for an uncached storage transfer unit on the at least one list with sufficient space (Fig 4: data partition portion; page 3 paragraph 45 maintains free space of a segment) and if such an uncached storage transfer unit can not be located, identifying an unallocated storage transfer unit from the at least one tail pointer (Fig 4: data partition portion; page 3 paragraph 45 detects a segment is full; page 5 paragraph 84 selects the unallocated segment).

As for claim 11, Rabii describes the step of maintaining at least one tail pointer to a plurality of contiguous unallocated storage transfer units. The claim rejected based on the same rationale as in the rejection claim 9. Rabbi further describes assigning objects into unallocated segments in page 5 paragraph 84. Thus in order to write objects into unallocated segments as taught, inherently some sort of pointer to point to the location of remaining unallocated segments in the segment ring is required, as shown in Rabbi's Fig 5.

As for claim 12, Rabii discloses wherein a cached copy of a storage transfer unit is written to the first storage medium in response to at least one of: (i) an object in the storage transfer unit being updated; (ii) a number of changed bytes in the storage transfer unit exceeding a threshold; (iii) a number of changed objects in the storage transfer unit exceeding a threshold; and (iv) the cached copy being about to be purged from the cache. Rabii describes the objects remain in cache until there is a shortage of in-memory-objects or data buffers. In this case the object is purged from cache, similar to item (iv) of the claim, that is the cached copy is being purged from cache because the cache resource is needed and reused for new data (Rabbi's page 7,paragraph 108, the cached copy of data transfer unit is written back to disk when the cache resources, data buffer, is reused for new data).

As for claim 14 the claim recites wherein the cached copy is about to be purged from the cache as a result of at least one of a cache replacement policy and the computer system being about to go down. The claim rejected based on the same rationale as the rejection of claim 12. Rabii further describes a large write back during a system shutdown process (Page 5, paragraph 80).

As for claim 15, Rabii describes wherein a storage transfer unit comprises a sector (Rabii page 3, paragraph 45, segment).

As for claim 16, the claim recites a method of maintaining a plurality of objects in a storage transfer unit, comprising the steps of: identifying an object position in the storage transfer unit by an object offset in the storage transfer unit; in response to a request to one of access and update a storage transfer unit, copying the storage transfer unit so that different objects are copied into different buffers; performing at least one update to at least one object in the storage transfer unit by modifying at least one buffer; and after the at least one update has occurred, updating the storage transfer unit from the at least one buffer. The claim rejected based on the same rationale as in the rejection of claim 1. Mattis describes the staging of data into buffers in memory and to write buffers back to the disk (Mattis's column 16, line 55 to column 17 line 2). Mattis further describes the arenas contain al least one fragment (corresponding to the claim's object; Mattis's column 17, lines 19-30).

As for claim 19, Rabii describes wherein the storage transfer unit is maintained on disk and the at least one buffer is maintained in main memory (Rabii's Fig 9, Fig 10).

As for claims 20-22, Rabii describes wherein the step of updating the storage transfer unit from the at least one buffer further comprises copying a plurality of objects from buffers to the storage transfer unit in a contiguous area so that free space in the storage transfer unit is contiguous (claim 20; Rabii's Fig 5: #240 free space, Rabbi's paragraph 79 discloses); wherein the storage transfer unit comprises a sector (claim 21; Rabii's Fig 5: #220 segment); the step of maintaining a number of free bytes in the storage transfer unit (claim 22; Rabii's Fig 5: #240 free space).

As for claim 23 Rabii describes wherein the step of performing at least one update further comprises using the number of free bytes in the storage transfer unit to prevent overflow (Rabii's paragraph 45, data structures to track the free blocks, and determine whether each segment is full).

As for claim 24, Rabii describes wherein the copy of at least one of the plurality of storage transfer units included in the cache is one of a partial copy and an inexact copy (Rabii's page 6, paragraph 106).

Claims 25, 27 rejected based on the same rationale as of claim 1,

Claims 26,28 rejected based on the same rationale as in the rejection of claim 16.

Claims 4-8,10 rejected under 35 U.S.C. 103(a) as being unpatentable over Rabii et al (US Pub 2002/0032691), Mattis et al (US 6915307) as applied to claims 1, and further in view of Chen et al (US 6804761).

As for claims 4-5, Rabbi does not describe the claim's giving the preference aspect while searching for a cached storage transfer unit further comprises identifying a cached storage transfer unit with sufficient free space to store the object giving preference to such cached storage transfer units with less free space; (claim 4); wherein the step of searching for a cached storage transfer unit further comprises identifying a cached storage transfer unit with a least amount of free space sufficient to store the object (claim 5). However, Chen describes a chunk manager capable of searching for the next highest size standard memory blocks to fits an application request (Chen's column 5, lines 1-20). It would have been obvious to one of ordinary skill in the art at the time of invention to include the block allocation method as suggested by Chen in Rabbi's system to reduce the fragmentation of memory (Chen's column 1, lines 60-64).

As for claims 6-8, the claims recite wherein the step of identifying an uncached storage transfer unit further comprises identifying an uncached storage transfer unit with sufficient free space giving preference to storage transfer units which minimize fragmentation (claim 6); wherein the step of identifying an uncached storage transfer unit further comprises giving preference to storage transfer units with more free space (claim 7); wherein the step of identifying an uncached storage transfer unit further comprises identifying a storage transfer unit with a most free space (claim 8). The claims rejected based on the same rationale as in the rejection of claims 4-5. Chen further describes of searching through a range of blocks with different sizes in order to determine an acceptable block and thereby reducing the fragmentation of memory space (Chen's column 5 line 65 to column 6 line 20). Examiner notes that the teaching of allocating free blocks, scanning through free blocks taught by Chen is applicable to memory spaces in any storage medium.

As for claim 10, the claim recites the step of maintaining a plurality of lists of storage transfer units organized by a quantity of free space in a storage transfer unit. It is rejected based on the same rejection of claim 6-8. Chen further describes memory pools with lists that organized based on free block sizes (Chen's column 5, pages 1-10; scanning memory pools having free block sizes 1K, 3K etc.. column 5, lines 43-55).

Claims 13,17 rejected under 35 U.S.C. 103(a) as being unpatentable over Rabii et al (US Pub 2002/0032691), Mattis et al (US 6915307) as applied to claims 1, and further in view of Cabrera et al (US 5802599).

As for claim 13, wherein an application program writes at least one storage transfer unit to disk in a transactional manner. Rabii does not describe the claim's aspect of writing in a transactional manner. However, Cabrera teaches that by providing writing objects in a way to reduce the fragmentation in the storage device, thus optimizing the access time to the device; The reducing in storage device access time would be advantageous for the system handling transactional requests such as database management system (Cabrera's column 4, lines 5-25). It would have been obvious to one of ordinary skill in the art at the time of invention to include the storage device accessing method as suggested by Cabrera in Rabbi's system to reduce the fragmentation of a storage device and thereby optimizing the access time to the storage device of the transactional based system such as database management system (Cabrera's column 4, lines 5-25).

As for claim 17, the claim recites when an object which does not have a highest offset is deleted, adding the offset to a list; satisfying an allocation request by using an offset from the list; and if an offset is not reused by the time the storage transfer unit is updated from the at least one buffer, storing a placeholder on the storage transfer unit indicating the object has been deleted. The claim rejected based on the same rationale as in the rejection of claim 1.Rabii further describes directory structures (Rabbi's Fig 8: #300 references field) to kept track of objects existing in memory. Rabii and Mattis do not describe the claim's aspect of reusing the memory buffer. However Carbrera describes the preferred steps of reusing the memory buffer (Carbrera's column 3, lines 40-47). It would have been obvious to one of ordinary skill in the art at the time of invention to include the objects grouping method as suggested by Carbrera in Rabbi's system to reduce the fragmentation of the storage device (Carbrera's column 3, lines 5-12).

Claim 18 rejected under 35 U.S.C. 103(a) as being unpatentable over Rabii et al (US Pub 2002/0032691), Mattis et al (US 6915307) as applied to claims 16, and further in view of Cabrera et al (US 5802599), Garthwaite (US Pub 2004/0172507).

As for claim 18, the claim recites in response to an object update which would cause a storage transfer unit to overflow, moving the object to a new storage transfer unit and storing a forwarding pointer in the previous storage transfer unit. The claim rejected based on the same rationale as in the rejection of claim 17. Rabii and Mattis do not describe the claim's aspect of overflowing. However, Cabrera further describes if the object does not fit in a used buffer containing one or more objects, that object is stored in a refreshed buffer (Cabrera's column 3, lines 40-47). It would have been obvious to one of ordinary skill in the art at the time of invention to include the objects grouping method as suggested by Cabrera in Rabbi's system to reduce the fragmentation of the storage device (Cabrera's column 3, lines 5-12). Rabii, Mattis and Cabrera do not describe the claim's detail of storing a forwarding pointer. However, Garthwaite describes a method of placing a forwarding pointer once an object has been moved (Garthwaite's page 7,.paragraph 63). It would have been obvious to one of ordinary skill in the art at the time of invention to include the forwarding pointer method as suggested by Garthwaite in Rabbi's system to allow processing the references contained in the object being evacuated while continuing of evacuating process (Garthwaite's Figs 10, 11A; page 7, paragraphs 63-65).

(10) Response to Argument

Appellant's arguments in response to the last office action has been fully considered but they are not persuasive. Examiner respectfully traverses Appellant's arguments for the following reasons:

Regarding Appellant's remarks at pages 9-13 for the rejections of claims 1-3, 9,11,12,14-28 under U.S.C. 103(a),

A) Appellant main argument at pages 10-12, is that Rabbi does not teach "a method of managing storage of <u>objects of size smaller than a storage transfer unit</u>". Examiner respectfully disagrees,

First, although Appellant's arguments refer to the sector of a disk as "the unit of storage transfer", there is no limitation in the claim that requires "the unit of storage transfer" to be a disk sector. In Addition, Appellant's specification on page 6 lines 9-12 defines sector" stating "Further, it is to be understood that the term "sector" as used herein refers not only to disk sectors but more generally **to any fixed unit** of transfer between two different storage media". Thus, according to Appellant's specification the claim's "storage transfer unit can reasonably be construed to read on a chunk or a unit of data of any size that is transferred between two storage medium, so long as the chunk or unit of data is fixed in size.. For example a data chunk such as one or more cache lines that are fetched from a cache memory and stored on a disk meets this definition.

Specifically, in Rabbi when needed data is read from the disk and then copied to a semiconductor RAM (cache) as discussed in paragraph 4, this claim limitation is met. In addition, in paragraphs 6, 7 and 12, Rabbi discloses cache lines corresponding to a file, a page or other data objects, such as a data segment storing one or more objections, that are transferred between the disk and memory cache, which also meets this claim limitation. Thus, Rabbi clearly discloses "storage transfer units" as claimed.

Second, Appellant argues that Rabbi does not teach the notion of an object smaller than a transfer unit. Examiner respectfully disagrees, Rabbi's paragraph 45, lines 6-13 further states "the data maintained should include enough information to determine whether each segment is full, **how many objects there are in the segment... ", which indicates that segments are made up of objects, thus objects must necessarily be smaller than segments (transfer units).** Thus Rabbi clearly objects that are smaller than transfer units.

B) Regarding Appellant's arguments at page 10, that Rabbi does not teach the claim's limitation "..maintaining a plurality of storage transfer units in a first storage medium organized by a quantity of free space..". Examiner disagrees. Rabbi discloses the organizing of free spaces in a storage transfer unit on page 3 in paragraphs 38-39 and as shown in Fig 5.

Appellant argues at page 10-11 that Rabbi does not teach the claim's limitation "...maintaining in a second storage medium a cache comprising a copy of at least one of said plurality of storage transfer units.." and "...no notion of storage transfers units maintained in the data buffer..". Examiner disagrees. Rabbi discloses a mechanism managing plural objects which are smaller and contained in a segment being transferred to/from a cache memory, and wherein the storage transfer unit can be any chunk of data, see discussion in item A.

Rabbi's paragraphs 102-104 further discloses data object that is stored in a memory buffer so that data can be quickly provided to the requestor (see paragraph 102, "memory object is created that is shared among all entities that access a given data object" and paragraph 104 "**the memory** object 500 holds a working set of one or more contiguous buffers..which are used to **hold the in-memory version of the on-disk data object 230**).

C) Appellant argues at page 11 that the combination of Rabbi and Mattis is improper. The argument is not persuasive. Examiner maintains that cited references are in fact analogous in that they relate to the field of managing the storage of data objects stored in memory such as a cache or disk. Pursuant to MPEP 2141.01(a), "in order to rely on a reference as a basis for rejection of an applicant's invention, the reference must either be in the field of applicant's endeavor or, if not then be reasonably pertinent to the particular problem with which the inventor was concerned..." In the instant case, the invention disclose by Mattis is in the data storage field as discussed in the abstract which states that the data objects are stored in cache and latter stored in persistent memory. The invention discloses by Rabbi is also in the data storage field as discussed in paragraphs 102-104 which state that data objects are stored in cache/memory buffers so that data can be quickly provided to the requestor. Moreover, both MAttis and Rabbi are directed to improving the availability and integrity of data stored in a storage system The examiner has properly combined the analogous prior art references, and further provided explicit motivation from each of the teaching references to properly establish why one of ordinary skill in the art would be sufficiently motivated to combine the teachings of Rabbi and Mattis. The rejections are therefore maintained since Examiner established a prima facie case of obviousness by properly asserting a motivation to combine these prior art references.

D) On page 12 with respect to the rejection of claim 16, Appellant repeats his arguments regarding the claim limitation "storage transfer unit". The Board's attention is respectfully directed to the Examiner's response to this argument set forth above. Appellant further contends that Rabbi fails to disclose "object offsets" and the claimed relationship between storage transfer units and object offsets. In Fig. 5, Rabbi discloses a segment (storage transfer unit) Since, multiple objects are stored in a segment these objects are necessarily arranged offset from each other within a segment (for example elements 230-1 and 230-2 are offset) .

Appellant further argues that Rabbi does not teach "copying a storage transfer unit in response to a request to one of access and update a storage transfer unit, copying the storage transfer unit so that different objects are copied into different buffer". Examiner disagrees. . Rabbi discloses that in response to a request to access the data, the data needed is read/retrieved from the disk and copied to a semiconductor RAM (paragraph 4, lines 15-18). Rabbi further discloses a set of different buffers to store objects in cache (each memory object has a different set of buffers 520). Retrieving data objects from disk to cache memory is easily done using the mapping of objects in the store directory table/information (paragraph 35 lines 4-8).

E) On page 12 with respect to claims 2 and 3, Appellant again argues that Rabbi fails to disclose any notion of an "object smaller than a storage transfer unit". As discussed above, the objects disclosed by Rabbi are smaller in size than the segments. F) On page 13 with respect to claims 9-11, Appellant continues to argue that Rabbi fails to disclose unallocated/free spaces of the disk. However, as discussed above, in Fig 5 and paragraph 84 Rabbi describes a mechanism for assigning objects into unallocated/free spaces of the disk. The free space locations are located at the tail end of other previously written objects in the disk. Furthermore when data is written into the free space locations at the end of a segment, a list of previous occupied object locations must be maintained. In other words, Rabbi clearly suggests limitations of using the list of storage transfer units and tail pointer pointing to the free/unallocated space in order to write the "uncached storage transfer unit" (data no longer being stored in cache) into the free space area of the disk storage media.

F) In response to Appellant's remarks at page 13, regarding the rejection of claim 12, In paragraph 108 Rabbi further discloses that the objects remain in cache until there is a shortage of in-memory-objects or data buffers. When the shortage of resources occurs, the objects are purged/destaged from the cache such that the resources can be reused/assigned to other objects/data. It's noted that the cache memory only store a portion of data being stored in the disk. Therefore, a cache replacement policy is required to stored/retrieve the proper data in cache to quickly provide the data to a requestor (paragraph 4).

Regarding Appellant's remarks at page 13 regarding the rejection of claim 14, the replacing of a cached entry based on a replacement policy is rejected based on the rationale as of claim 12 above and as discussed in item F above. Rabbi further discloses the write back to disk can be accomplished in one large raw write operation in a rapid fashion (paragraph 80, lines 1-5) It is quick and efficient, because the data is transferred and stored in one contiguous disk segment in case of an unexpected failure (paragraph 86, lines 1-8) for example flushing cache when power failure occurs. In other words, Rabbi clearly teaches that in case of unexpected failure (power down), the cached copy of data in cache (data managed by a cache replacement policy) must be purged quickly to disk so that data can be safely stored in disk/permanent storage device before losing power to the system.

Regarding Appellant's remarks at pages 13-14 for the rejection of claims 19 and 20-24, the claim terminology "storage transfer units" and "object offsets" are addressed in above paragraphs, specifically items A,B and D above.

G) Regarding Appellant's remarks at pages 13-14 for the rejection of claims 4-8, and 10 under U.S.C 103, managing of the objects with a size smaller than a segment is rejected as discussed in item A above. The remaining claim limitations associated with free space (free area of a segment not being assigned to any object) is clearly shown in the method disclosed by Chen. Examiner maintains that Rabbi and Mattis are properly combinable as discussed in above items A,B and C. And the examiner maintains that there is proper motivation for combing the teachings of Chen with Rabbi's system as modified by Mattis as follows:

Although Rabbi and Mattis do not expressly discloses the claim's details with respect to searching, scanning, and determining free blocks, Chen clearly discloses a chunk manager capable of searching, scanning and determining free block as recited in the claims (see the rejections for claims 4-8,10). It would have been obvious to one of ordinary skill in the art at the time of invention to include the block allocation method as suggested by Chen in Rabbi's system modified by Mattis to further reduce the fragmentation of memory (Chen's column 1, lines 60-64).

H) Appellant argues on page 14 with respect to the rejection of claims 13 and 17 that the references fail to disclose "objects stored in an offset manner,

The limitation of "objects are stored in offset manner" is discussed in items A,B and D above. Examiner further maintains that the combination and motivation for combing Rabbi and Mattis is proper as discussed in item C above. The examiner further maintains that the motivation for combing Cabrera's teachings in Rabbi's system as modified by Mattis is proper as discussed in the rejection of claims 13, 17.

I) Regarding Appellant's remarks at page 14 for the rejection of claim 18,

Examiner maintains that Cabrera further discloses that if the object does not fit in a used buffer which containing one or more objects, that object is stored in a refreshed buffer (Cabrera's column 3, lines 40-47).

Regarding Appellant's argues that Garthwaite does not teach the claim's limitation of storing a forwarding pointer. Examiner disagrees. Garthwaite further discloses a method of placing a forwarding pointer once an object has been moved (Garthwaite's page 7, paragraph 63).. Examiner further maintains that the combination and motivation of references Rabbi, Mattis, Cabrera is proper as discussed in above items C, H and the motivation for combination of Garthwaite's teaching in Rabbi's system modified by Mattis and Cabrera is proper as discussed in the rationale of claim 18.

(11) Related Proceeding(s) Appendix

No decision rendered by a court or the Board is identified by the examiner in the Related Appeals and Interferences section of this examiner's answer.

For the above reasons, it is believed that the rejections should be sustained.

Respectfully submitted,

/Duc T Doan/

Duc Doan
Patent Examiner
Art Unit 2188

Conferees:

/Lynne H Browne/
Lynne H Browne
Appeal Practice Specialist, TQAS
Technology Center 2100

HYUNG S. SOUGH
SUPERVISORY PATENT EXAMINER
11/30/07